



## Espacenet

**Bibliographic data:** JP 2004038977 (A)

# IMAGE FILE MANAGEMENT METHOD AND RECORDING MEDIUM THEREFOR

**Publication date:** 2004-02-05

**Inventor(s):** WITT WESLEY A; SCHUMACHER SARA J; RAGHAVEN KARTIK N +

**Applicant(s):** MICROSOFT CORP +

**Classification:**

- international: *G06F12/00; G06F12/08; G06F12/14; G06F15/16; G06F17/30; (IPC1-7): G06F12/00; G06F12/14*

- European: *G06F11/14A4B1M6*

**Application number:** JP20030185958 20030627

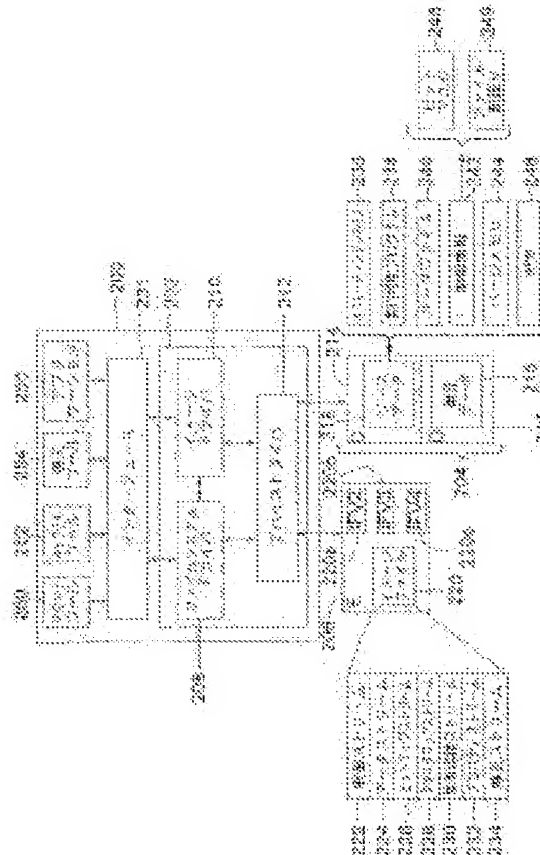
**Priority number (s):** US20020184257 20020628

**Also published as:**

- JP 4406224 (B2)
- EP 1376404 (A2)
- EP 1376404 (A3)
- US 2007112826 (A1)
- US 7877567 (E2)
- more

Abstract of JP 2004038977 (A)

**PROBLEM TO BE SOLVED:** To provide an image file management method and a recording medium therefor. ; **SOLUTION:** An image file 220 comprises a plurality of streams including a control stream 222, a data stream 224, a bitmap stream 226, and a cluster stream 228. In addition, an audit trail stream 130, a property stream 232, and a modification stream 234 can also be provided. An image driver 210 can read and edit an image by converting a request from an operating system 236, or can operate the image using other methods. ; **COPYRIGHT:** (C)2004, JPO



Last updated:  
 26.04.2011 Worldwide  
 Database 5.7.23.1; 93p

(19) 日本国特許庁 (JP)

(12) 公 開 特 許 公 報 (A)

(11) 特許出願公開番号

特開2004-38977

(P2004-38977A)

(43) 公開日 平成16年2月5日 (2004.2.5)

(51) Int. Cl. <sup>7</sup>

G06F 12/00

G06F 12/14

F 1

G06F 12/00 531M

G06F 12/00 511A

G06F 12/00 537H

G06F 12/14 320B

テーマコード (参考)

5B017

5B082

審査請求 未請求 請求項の数 30 O L (全 25 頁)

(21) 出願番号 特願2003-185958 (P2003-185958)

(22) 出願日 平成15年6月27日 (2003.6.27)

(31) 優先権主張番号 10/184, 257

(32) 優先日 平成14年6月28日 (2002.6.28)

(33) 優先権主張国 米国 (US)

(71) 出願人 391055933

マイクロソフト コーポレーション

MICROSOFT CORPORATI  
ON

アメリカ合衆国 ワシントン州 9805  
2-6399 レッドモンド ワン マイ  
クロソフト ウェイ (番地なし)

(74) 代理人 100077481

弁理士 谷 義一

(74) 代理人 100088915

弁理士 阿部 和夫

(72) 発明者 ウェスリー エー. ウィット

アメリカ合衆国 98053 ワシントン  
州 レッドモンド ノースイースト 12  
2 ストリート 20935

最終頁に続く

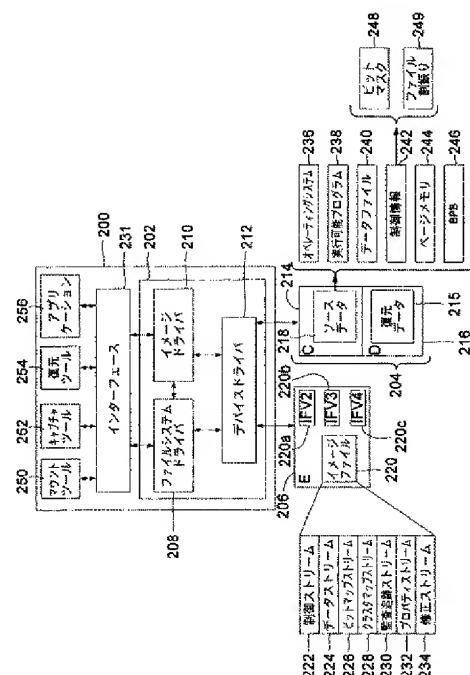
(54) 【発明の名称】 イメージファイル管理方法及びその記録媒体

(57) 【要約】

【課題】イメージファイル管理方法及びその記録媒体を提供すること。

【解決手段】イメージファイル220が、制御ストリーム222とデータストリーム224とビットマップストリーム226とクラスタマップストリーム228などの複数のストリームを備えている。また、監査追跡ストリーム130とプロパティストリーム232と修正ストリーム234も提供されることが可能である。イメージドライバ210が、オペレーティングシステム236からの要求を変換して、イメージを読み取ること及び編集すること、または他の方法で操作することが可能であるようにする。

【選択図】 図2



**【特許請求の範囲】****【請求項1】**

データを含むデータストリーム、ならびに前記データの仮想の場所、および前記データのデータストリーム域をそれぞれが特定する複数のレコードを含むクラスタマップストリームを含むイメージファイル内の前記データにアクセスするためのイメージファイル管理方法であって、

前記データの仮想の場所を特定するアクセス要求を受け取るステップと、  
前記複数のレコードの1つを使用して前記仮想の場所を前記データストリーム域にマップするステップと、

前記データストリームから前記データを読み取るステップと  
を備えたことを特徴とするイメージファイル管理方法。

**【請求項2】**

前記アクセス要求が、アプリケーションプログラムから開始され、かつ前記データが、前記アプリケーションプログラムによって受け取られることを特徴とする請求項1に記載のイメージファイル管理方法。

**【請求項3】**

前記アクセス要求が、ユーザインターフェースによって開始されることを特徴とする請求項1に記載のイメージファイル管理方法。

**【請求項4】**

前記アクセス要求が、コマンドライン要求によって開始されることを特徴とする請求項1に記載のイメージファイル管理方法。

**【請求項5】**

前記データを編集するステップと、  
前記編集されたデータを前記イメージファイル内の前記データストリームに書き込むステップと  
を備えたことを特徴とする請求項1に記載のイメージファイル管理方法。

**【請求項6】**

前記編集されたデータが、前記データストリームの終端に書き込まれることを特徴とする請求項5に記載のイメージファイル管理方法。

**【請求項7】**

前記データにアプリケーションプログラムからアクセスするステップと、  
前記データを編集するステップと、  
前記編集されたデータを前記イメージファイル内の前記データストリームに書き込むステップと  
を備えたことを特徴とする請求項1に記載のイメージファイル管理方法。

**【請求項8】**

前記編集されたデータが、前記データストリームの終端に書き込まれることを特徴とする請求項7に記載のイメージファイル管理方法。

**【請求項9】**

前記仮想の場所は、1組のクラスタを含むことを特徴とする請求項1に記載のイメージファイル管理方法。

**【請求項10】**

前記クラスタは、連続していることを特徴とする請求項9に記載のイメージファイル管理方法。

**【請求項11】**

前記データストリーム内の前記データが圧縮されるイメージファイル管理方法であって、  
前記データを圧縮解除するステップを備えたことを特徴とする請求項1に記載のイメージファイル管理方法。

**【請求項12】**

前記データストリーム内の前記データが暗号化されるイメージファイル管理方法であって、前記データを暗号化解除するステップを備えたことを特徴とする請求項1に記載のイメージファイル管理方法。

【請求項13】

マウント要求を受け取るステップと、アクセスのために前記イメージファイルを開くステップと、前記イメージファイルに対する仮想ドライブマウントポイントを提供するステップとを備え、前記アクセス要求は、前記仮想ドライブマウントポイントを有することを特徴とする請求項1に記載のイメージファイル管理方法。

【請求項14】

データストリームおよびクラスタマップストリームを含む複数のストリームを含む第1のイメージファイルを編集するイメージファイル管理方法であって、イメージドライブにおいて、1組のデータを前記第1のイメージファイルの中に記憶する要求を受け取り、前記要求は、前記1組のデータに関する仮想の場所を含むステップと、前記1組のデータを前記データストリーム内の場所に追加するステップと、前記1組のデータに関する前記仮想の場所、および前記1組のデータに関する前記データストリーム内の前記場所を含むクラスタマップレコードを生成するステップと、前記クラスタマップレコードを前記クラスタマップストリームに追加するステップとを備えたことを特徴とするイメージファイル管理方法。

【請求項15】

前記1組のデータが、前記データストリームの終端に追加されることを特徴とする請求項14に記載のイメージファイル管理方法。

【請求項16】

前記1組のデータに関する前記仮想の場所が、割振り済みクラスタおよび未割振りクラスタを特定するビットマスクストリームから決定され、前記仮想の場所は、前記未割振りクラスタの1つまたは複数を含むことを特徴とする請求項14に記載のイメージファイル管理方法。

【請求項17】

前記イメージファイルは、監査追跡ストリームを有するイメージファイル管理方法であって、前記データストリームに追加される前記1組のデータを含むファイルを特定する監査追跡レコードを生成するステップを備えたことを特徴とする請求項14に記載のイメージファイル管理方法。

【請求項18】

制御ストリーム、データストリーム、およびクラスタマップストリームを含むイメージファイルを少なくとも第1のコンピュータおよび第2のコンピュータを含むネットワークを介して伝送するイメージファイル管理方法であって、前記イメージファイルを構文解析して、前記制御ストリーム、前記クラスタマップストリーム、および前記データストリームの第1の部分を構成する第1のセグメントと、前記データストリームの第2の部分を構成する第2のセグメントとを少なくとも含む複数のセグメントに分けるステップと、前記第1のセグメントを前記第1のコンピュータから前記第2のコンピュータに送信するステップと、前記第2のセグメントを前記第1のコンピュータから前記第2のコンピュータに送信するステップとを備えたことを特徴とするイメージファイル管理方法。

【請求項19】

前記第1のセグメントおよび前記第2のセグメントは、実質的に同じ長さであることを特徴とする請求項18に記載のイメージファイル管理方法。

【請求項20】

前記長さは、ユーザによる選択が可能であることを特徴とする請求項19に記載のイメージファイル管理方法。

**【請求項21】**

コンピュータにおいて、ソース記憶媒体上のソースデータから複数のイメージファイルを作成するイメージファイル管理方法であって、  
データの前記ソースを含む第1のデータストリーム、固有識別子を有する第1の制御ストリーム、および第1のクラスタマップストリームを含む第1のイメージファイルを前記ソースデータから作成し、前記第1のイメージファイルを宛先記憶媒体ボリューム上に記憶するステップと、  
前記ソース記憶媒体ボリューム上の前記ソースデータを変更するステップと、前記変更されたソースデータを含む第2のデータストリーム、前記第1のイメージファイルの前記固有識別子をポイントする制御ストリーム、および第2のクラスタマップストリームを含む第2のイメージファイルを前記変更された1組のデータから作成し、前記第2のイメージファイルを前記宛先記憶媒体ボリューム上に記憶するステップと  
を備えたことを特徴とするイメージファイル管理方法。

**【請求項22】**

前記第1のイメージファイルおよび前記第2のイメージファイルが、単一のファイルの一部として記憶されることを特徴とする請求項21に記載のイメージファイル管理方法。

**【請求項23】**

所望のデータを含むデータストリーム、ならびに前記所望のデータの仮想の場所、および前記所望のデータのデータストリーム域をそれぞれが特定する複数のレコードを含むクラスタマップストリームを含む複数のストリームを含むイメージファイル内の前記所望のデータにアクセスするためのイメージファイル管理方法のためのコンピュータ実行可能命令を有するコンピュータ読み取り可能な記録媒体であって、  
前記所望のデータの前記仮想の場所を特定するアクセス要求を受け取るステップと、  
前記複数のレコードの1つを使用して前記仮想の場所を前記データストリーム域に変換するステップと、  
前記データストリームから前記データを読み取るステップと  
を備えたことを特徴とするコンピュータ読み取り可能な記録媒体。

**【請求項24】**

前記データストリーム内の前記所望のデータが圧縮されるコンピュータ読み取り可能な記録媒体であって、前記所望のデータを圧縮解除するステップを備えたことを特徴とする請求項23に記載のコンピュータ読み取り可能な記録媒体。

**【請求項25】**

前記データストリーム内の前記所望のデータが暗号化されるコンピュータ読み取り可能な記録媒体であって、前記所望のデータを暗号化解除するステップを備えたことを特徴とする請求項23に記載のコンピュータ読み取り可能な記録媒体。

**【請求項26】**

マウント要求を受け取るステップと、アクセスのために前記イメージファイルを開くステップと、前記イメージファイルに対する仮想ドライブマウントポイントを提供するステップとを備え、前記アクセス要求は、前記仮想ドライブマウントポイントを含むことを特徴とする請求項23に記載のコンピュータ読み取り可能な記録媒体。

**【請求項27】**

データストリームおよびクラスタマップストリームを含む複数のストリームを含む第1のイメージファイルを編集するイメージファイル管理方法のためのコンピュータ実行可能命令を有するコンピュータ読み取り可能な記録媒体であって、  
イメージドライバにおいて、1組のデータを前記第1のイメージファイルの中に記憶する要求を受け取り、前記要求は、前記1組のデータに関する仮想の場所を含むステップと、  
前記1組のデータを前記データストリーム内の場所に追加するステップと、  
前記1組のデータに関する前記仮想の場所、および前記1組のデータに関する前記データストリーム内の前記場所を含むクラスタマップレコードを生成するステップと、  
前記クラスタマップレコードを前記クラスタマップストリームに追加するステップと

を備えたことを特徴とするコンピュータ読み取り可能な記録媒体。

【請求項28】

前記イメージファイルは、監査追跡ストリームを有するコンピュータ読み取り可能な記録媒体であって、前記データストリームに追加される前記1組のデータを含むファイルを特定する監査追跡レコードを生成するステップを備えたことを特徴とする請求項27に記載のコンピュータ読み取り可能な記録媒体。

【請求項29】

ソース記憶媒体上のソースデータから複数のイメージファイルを作成するイメージファイル管理方法のためのコンピュータ実行可能命令を有するコンピュータ読み取り可能な記録媒体であって、

データの前記ソースを含む第1のデータストリーム、固有識別子を有する第1の制御ストリーム、および第1のクラスタマップストリームを含む第1のイメージファイルを前記ソースデータから作成し、前記第1のイメージファイルを宛先記憶媒体ボリューム上に記憶するステップと、

前記ソース記憶媒体ボリューム上の前記ソースデータを変更するステップと、前記変更されたソースデータを含む第2のデータストリーム、前記第1のイメージファイルの前記固有識別子をポイントする制御ストリーム、および第2のクラスタマップストリームを含む第2のイメージファイルを前記変更された1組のデータから作成し、前記第2のイメージファイルを前記宛先記憶媒体ボリューム上に記憶するステップと

を備えたことを特徴とするコンピュータ読み取り可能な記録媒体。

【請求項30】

1組のデータをそれぞれが含む複数のストリームを含むイメージファイルからシステムツールが情報を獲得する方法であって、

インターフェースにおいて、前記複数のストリームの少なくとも1つから前記1組のデータの要求を受け取るステップと、

前記インターフェースから、オペレーティングシステム内で実行されるイメージドライバに前記要求を送るステップと、

前記複数のストリームの前記少なくとも1つから前記1組のデータを読み取るステップと、

前記複数のストリームの前記少なくとも1つからの前記1組のデータを前記システムツールに提供するステップと

を備えたことを特徴とする方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、イメージファイル管理方法及びその記録媒体に関し、より詳細には、イメージファイルを管理するコンピュータシステムに関する。

【0002】

【従来の技術】

コンピュータシステムは、1つまたは複数の記憶媒体（記録媒体）とインターフェースとを備えている。記憶媒体は、オペレーティングシステムファイル、アプリケーションプログラム、およびアプリケーションプログラムで使用されるデータファイルなどの様々なデータを記憶する。周知の記憶媒体のタイプには、ハードディスク、CD-ROM、デジタルバーサタイルディスク（DVD）等が含まれる。記憶媒体は、1つまたは複数のボリューム、つまりパーティションに論理的に細分される。記憶媒体は、複数のセクタにさらに物理的に細分される。各セクタは、複数のデータバイトを記憶することができる。クラスタは、一群のセクタであり、コンピュータで実行されているオペレーティングシステムが記憶媒体上の場所を特定するのに使用する最小単位を表わす。したがって、オペレーティングシステムは、通常、クラスタごとにクラスタ上で記憶媒体上のデータの記憶、または読取りを行う。

【0003】

記憶媒体上のデータは、1組のファイル、およびディスク上のファイルを管理するのに使用される他の制御情報として編成される。例えば、各オペレーティングシステムファイル、アプリケーションプログラム、またはデータファイルは、記憶媒体上の異なるファイルを表わす。制御情報は、各ファイルに関するデータを含む記憶媒体上のクラスタを特定する。また、制御情報は、割り振られた、すなわち、ファイルに関するデータを含む記憶媒体上のクラスタ、および割り振られていないままの、すなわち、新しいデータを記憶するのに利用可能なクラスタも特定する。制御情報自体、1つまたは複数のファイルの一部として記憶される。ファイルおよび制御情報が記憶媒体上で厳密にどのように編成されるかは、ファイルシステムに依存する。ファイルアロケーションテーブル (File Allocation Table) 16 (FAT16)、ファイルアロケーションテーブル32 (FAT32)、およびニューテクノロジーファイルシステム (New Technology File System) (NTFS) などの様々な周知のファイルシステムが存在する。

#### 【0004】

イメージファイルは、ソース記憶媒体ボリューム上に記憶されたデータのコピーである。通常、イメージファイルは、ソース記憶媒体ボリュームに含まれるデータのセクタごとのコピーである単一のデータストリームである。イメージファイルは、宛先記憶媒体ボリューム上に記憶される。宛先記憶媒体ボリュームは、同一の記憶媒体上の異なるボリューム、つまり異なるパーティションであるか、あるいは別個の記憶媒体である。例えば、ハードディスクボリュームのセクタごとのコピーを表わすイメージファイルが、CD-ROM上に記憶される。

#### 【0005】

イメージファイルは、少なくとも2つの機能のために使用される。第1に、イメージファイルは、ソース記憶媒体ボリュームをイメージファイルが作成された時点の状態に復元するのに使用される。したがって、記憶媒体ボリュームが損傷した場合、または壊れた場合、イメージファイルを使用してその記憶媒体ボリュームを復元することができる。第2に、イメージファイルは、複数のコンピュータシステムに同一の基本構成を提供するのに使用される。

#### 【0006】

##### 【発明が解決しようとする課題】

しかしながら、イメージファイルは、ソース記憶媒体のセクタごとのコピーとして作成され、単一のデータストリームとして記憶されるため、イメージファイルを操作することは困難である。オペレーティングシステムは、データストリームから制御情報または個々のファイルデータを容易に読み取ることができない。この結果、データストリーム内のファイルを新しいファイルで置き換えること、または編集することができない。イメージファイルを更新するため、イメージファイルを記憶媒体ボリュームに復元し、記憶媒体ボリュームに更新を行った後、新しいイメージファイルを作成することが必要である。また、ユーザまたはシステム管理者が異なるバージョンのイメージファイルを保持することを望む場合、それぞれの異なるバージョンに関して別個のイメージファイルを作成しなければならない。したがって、イメージファイルを保持する周知の方法は、リソースを多く使用し、高い費用がかかる。

#### 【0007】

本発明は、このような問題に鑑みてなされたもので、その目的とするところは、イメージファイル管理方法及びその記録媒体を提供することにある。

#### 【0008】

##### 【課題を解決するための手段】

前述したように、本発明は、イメージファイル管理方法及びその記録媒体を提供するものである。イメージファイルは、ソース記憶媒体ボリューム上のソースデータから作成される。イメージファイルは、同一のコンピュータと、または、好ましくは、LAN、WAN、またはインターネットなどのネットワークを介して通信する他のコンピュータとインタ

ーフェースをとる宛先記憶媒体ボリューム上に記憶される。イメージファイルは、複数のイメージストリームを含む。イメージストリームには、制御ストリーム、データストリーム、ビットマップストリーム、およびクラスタマップストリームが含まれる。監査追跡ストリーム、プロパティストリーム、および修正ストリームなどのその他のイメージストリームが、オプションとしてイメージファイルの中に含まれる。

【0009】

制御ストリームは、イメージファイルを一意的に特定するイメージファイルのヘッダを形成する。また制御ストリームは、ソース記憶媒体ボリュームジオメトリ ( geometry )、およびソース記憶媒体上に記憶されたオペレーティングシステムを特定するデータなどの、ソース記憶媒体ボリュームを特定するデータも含む。制御ストリームにより、イメージファイルを読み取る、復元する、編集する、または別の仕方で作成するのに必要な情報が提供される。

【0010】

データストリームは、ソース記憶媒体ボリューム上の複数のファイルのそれぞれに関するデータを含む。また、データストリームは、ソース記憶媒体ボリュームからの制御情報も含む。データストリーム内のデータは、オプションとして、圧縮されるか、または暗号化される。本発明の実施形態では、データストリーム内のデータは、イメージファイルが作成されたときにデフラグ ( defragment ) される。

【0011】

ビットマップストリームは、ファイルに割り振られているソース記憶媒体上のクラスタ、および割り振られていないままのクラスタを特定する。ビットマップストリームは、オペレーティングシステムが新しいデータを記憶するのに利用可能なソース記憶媒体上の場所を特定するのを可能にする迅速なルックアップを提供することにより、イメージファイルを編集することを容易にする。

【0012】

クラスタマップストリームは、ソース記憶媒体上の一群のデータの場所をデータストリーム内の場所に相関させる。プロパティストリームや監査追跡ストリームなどのオプションのイメージファイルストリームは、イメージファイル内にあるファイルのタイプを特定し、またイメージファイルに行われた編集を特定する。

【0013】

イメージドライバは、オペレーティングシステムがイメージファイルにアクセスするのを可能にするインターフェースを提供する。イメージドライバは、仮想クラスタ域 ( virtual cluster location ) を含むアクセス要求をデータストリーム域に変換する。イメージドライバは、あたかもイメージファイルが別個にアドレス指定可能な記憶媒体ボリュームであるかのように、オペレーティングシステムがイメージファイルを読み取ることを可能にする。

【0014】

イメージドライバは、イメージファイルの編集を可能にする。イメージドライバは、データをイメージファイルに書き込む要求を受け取ったとき、そのデータをデータストリームの終端に書き込む。イメージドライバは、仮想クラスタ域をデータストリーム域に相関させるクラスタマップレコードを提供することによってクラスタマップストリームを更新する。

【0015】

【発明の実施の形態】

以下、図面を参照して本発明の実施例について説明する。

本発明は、イメージファイル管理方法及びそのシステム ( 記録媒体 ) を対象としている。イメージファイルは、ソース記憶媒体ボリューム上に記憶されたデータのコピーを含む。ソース記憶媒体ボリューム上に記憶されるデータのタイプには、通常、オペレーティングシステムファイル、アプリケーションプログラム、データファイル、およびボリューム上に記憶されたデータを管理するのに使用されるその他の制御情報が含まれる。本発明では



、イメージファイルは、制御ストリーム、データストリーム、ビットマップストリーム、修正ストリーム、およびクラスタマップストリームを含む複数のストリームを含む。プロパティストリームや監査追跡などのオプションのストリームもイメージファイルの中を含めることができる。

【0016】

イメージファイルのデータストリームは、ソース記憶媒体ボリュームからコピーされたデータを含む。制御ストリーム、ビットマップストリーム、およびクラスタマップストリームは、データストリーム内のデータを管理するのに必要とされる情報を含む。イメージファイルには、イメージファイルに仮想記憶媒体リファレンスを提供するマウントプロセスによってアクセスが行われる。イメージファイルがマウントされた後、オペレーティングシステムが、仮想クラスタを参照することによってデータストリーム内のデータにアクセスする。クラスタマップストリームは、仮想クラスタをデータストリーム内の場所に変換するのに使用される。本発明により、あたかもイメージファイルがディスクドライブなどの記憶媒体ボリュームであるかのように、オペレーティングシステムがイメージファイルにアクセスすることが可能になる。例えば、データストリーム内に記憶されたファイルの閲覧、編集、作成、または削除を行うことができる。

【0017】

本発明の実施形態では、複数のバージョンのソース記憶媒体ボリュームが、1つまたは複数のイメージファイルの中に記憶される。親イメージファイルと呼ばれる第1のイメージファイルが、第1の期間にソース記憶媒体ボリュームから作成される。子イメージファイルと呼ばれる第2のイメージファイルが、第2の期間にソース記憶媒体ボリュームから作成される。子イメージファイルは、第1の期間以来、変更されているソース記憶媒体ボリュームからのデータだけを含む。第1のイメージファイルおよび第2のイメージファイルは、別個のファイルとして記憶される。別法では、第1のイメージファイルおよび第2のイメージファイルは、単一のイメージファイルの一部として記憶される。第1のイメージファイルおよび第2のイメージファイルにより、宛先記憶媒体ボリュームに、第1の期間、または第2の期間でソース記憶媒体ボリュームからのデータが提供されることが可能になる。

【0018】

各図面において、同じ構成要素には同一の符号が付けられており、本発明が適切な計算環境において実施されているものとして示されている。必須ではないが、本発明は、パーソナルコンピュータによって実行される、プログラムモジュールなどの、コンピュータ実行可能命令の一般的な文脈で説明する。一般に、プログラムモジュールには、特定のタスクを行う、または特定の抽象データタイプを実装するルーチン、プログラム、オブジェクト、構成要素、データ構造等が含まれる。また、本発明は、タスクが、通信網を介してリンクされた遠隔の処理デバイスによって行われる分散計算環境において実施することも可能である。分散計算環境では、プログラムモジュールは、ローカルのメモリ記憶デバイスと遠隔のメモリ記憶デバイスの両方の中に配置されることが可能である。

【0019】

図1は、本発明を実施することが可能な適切な計算システム環境の例を示す図である。計算システム環境100は、適切な計算環境の一例に過ぎず、本発明の使用または機能の範囲について全く限定を示唆するものではない。また、計算環境100が、例示的な動作環境100に示した構成要素のいずれか1つ、またはいずれかの組合せに関連する何らかの依存性または要件を有するものと解釈してはならない。

【0020】

本発明は、多数の他の汎用または特殊目的の計算システム環境または計算システム構成で動作する。本発明で使用するのに適する可能性がある周知の計算システム、計算環境、および／または計算構成の例には、パーソナルコンピュータ、サーバコンピュータ、ハンドヘルドデバイスまたはラップトップデバイス、マルチプロセッサシステム、マイクロプロセッサベースのシステム、セットトップボックス、プログラマブル家庭用電化製品、ネッ

トワークPC、ミニコンピュータ、メインフレームコンピュータ、以上のシステムまたはデバイスの任意のものを含む分散計算環境等が含まれるが、以上には限定されない。

【0021】

本発明は、コンピュータによって実行される、プログラムモジュールなどのコンピュータ実行可能命令の一般的な文脈で説明することが可能である。一般に、プログラムモジュールには、特定のタスクを行う、または特定の抽象データタイプを実装するルーチン、プログラム、オブジェクト、構成要素、データ構造等が含まれる。また、本発明は、タスクが、通信網を介してリンクされた遠隔の処理デバイスによって行われる分散計算環境において実施することも可能である。分散計算環境では、プログラムモジュールは、メモリ記憶デバイスを含むローカルのコンピュータ記憶媒体と遠隔のコンピュータ記憶媒体の両方の中に配置されることが可能である。

【0022】

図1を参照すると、本発明を実施するための例示的なシステムが、コンピュータ110の形態で汎用計算デバイスを含んでいる。コンピュータ110の構成要素には、処理ユニット120、システムメモリ130、ならびにシステムメモリを含む様々なシステム構成要素を処理ユニット120に結合するシステムバス121が含まれる可能性があるが、以上には限定されない。システムバス121は、様々なバスアーキテクチャの任意のものを使用するメモリバスまたはメモリコントローラ、周辺バス、およびローカルバスを含むいくつかのタイプのバス構造の任意のものであることが可能である。例として、限定としてではなく、そのようなアーキテクチャには、インダストリスタンダードアーキテクチャ(Industry Standard Architecture)(ISA)バス、マイクロチャネルアーキテクチャ(Micro Channel Architecture)(MCA)バス、エンハンスドISA(Enhanced ISA)(EISA)バス、ビデオエレクトロニクススタンダーズアソシエート(Video Electronics Standards Associate)(VESA)ローカルバス、およびメザニン(Mezzanine)バスとしても知られるペリフェラルコンポーネントインターコネクト(Peripheral Component Interconnect)(PCI)バスが含まれる。

【0023】

コンピュータ110は、通常、様々なコンピュータ読み取り可能な記録媒体を含む。このコンピュータ読み取り可能な記録媒体は、コンピュータ110によってアクセス可能な任意の利用可能な媒体であることが可能であり、揮発性の媒体と不揮発性の媒体、取外し可能な媒体と取外し不可能な媒体がともに含まれる。例として、限定としてではなく、コンピュータ読み取り可能な記録媒体は、コンピュータ記憶媒体および通信媒体を含むことが可能である。コンピュータ記憶媒体には、コンピュータ可読命令、データ構造、プログラムモジュール、またはその他のデータなどの情報を記憶するための任意の方法または任意の技術で実装される揮発性の媒体および不揮発性の媒体、取外し可能な媒体および取外し不可能な媒体がともに含まれる。コンピュータ記憶媒体には、RAM、ROM、EEPROM、フラッシュメモリまたは他のメモリ技術、CD-ROM、デジタルバーサタイルディスク(DVD)または他の光ディスクストレージ、磁気カセット、磁気テープ、磁気ディスクストレージまたは他の磁気記憶デバイス、あるいは所望の情報を記憶するのに使用することができ、コンピュータ110がアクセスすることができる任意の他の媒体が含まれるが、以上には限定されない。通信媒体は、通常、搬送波などの変調されたデータ信号、または他のトランスポート機構でコンピュータ可読命令、データ構造、プログラムモジュール、またはその他のデータを実体化し、あらゆる情報配信媒体が含まれる。「変調されたデータ信号」という用語は、信号に情報を符号化するような仕方で特性の1つまたは複数が設定または変更されている信号を意味する。例として、限定としてではなく、通信媒体には、有線ネットワークまたは直接配線接続などの有線媒体、ならびに音響媒体、RF媒体、赤外線媒体、およびその他の無線媒体などの無線媒体が含まれる。また、前述したもののいずれかの組合せも、コンピュータ読み取り可能な記録媒体の範囲に含まれ

るべきものである。

【0024】

システムメモリ130は、読取り専用メモリ（ROM）131およびランダムアクセスメモリ（RAM）132などの揮発性メモリおよび／または不揮発性メモリの形態でコンピュータ記憶媒体を含む。始動中など、コンピュータ110内部の要素間で情報を転送するのを助ける基本ルーチンを含む基本入力／出力システム133（BIOS）が、通常、ROM131の中に記憶されている。RAM132は、通常、処理ユニット120によって即時にアクセス可能であり、かつ／または現在、処理されているデータおよび／またはプログラムモジュールを含む。例として、限定としてではなく、図1は、オペレーティングシステム134、アプリケーションプログラム135、その他のプログラムモジュール136、およびプログラムデータ137を示している。

【0025】

また、コンピュータ110は、他の取外し可能な／取外し不可能な、揮発性／不揮発性のコンピュータ記憶媒体を含むことも可能である。単に例として、図1は、取外し不可能な不揮発性の磁気媒体に対して読取りおよび書込みを行うハードディスクドライブ141、取外し可能な不揮発性の磁気ディスク152に対して読取りおよび書込みを行う磁気ディスクドライブ151、およびCD-ROMまたは他の光媒体などの取外し可能な不揮発性の光ディスク156に対して読取りおよび書込みを行う光ディスクドライブ155を示している。例示的な動作環境において使用することができる他の取外し可能な／取外し不可能な揮発性／不揮発性のコンピュータ記憶媒体には、磁気テープカセット、フラッシュメモリカード、デジタルバーサタイルディスク、デジタルビデオテープ、ソリッドステートRAM、ソリッドステートROM等が含まれるが、以上には限定されない。ハードディスクドライブ141は、通常、インターフェース140のような取外し不可能なメモリインターフェースを介してシステムバス121に接続され、また磁気ディスクドライブ151および光ディスクドライブ155は、通常、インターフェース150のような取外し可能なメモリインターフェースでシステムバス121に接続される。

【0026】

前述し、図1に示すドライブおよび関連するコンピュータ記憶媒体により、コンピュータ可読命令、データ構造、プログラムモジュール、およびその他のデータのストレージがコンピュータ110に提供される。図1では、例えば、ハードディスクドライブ141が、オペレーティングシステム144、アプリケーションプログラム145、他のプログラムモジュール146、およびプログラムデータ147を記憶しているものとして示されている。以上の構成要素は、オペレーティングシステム134、アプリケーションプログラム135、他のプログラムモジュール136、およびプログラムデータ137と同じであることも、異なることも可能であることに留意されたい。オペレーティングシステム144、アプリケーションプログラム145、他のプログラムモジュール146、およびプログラムデータ147に、ここでは、少なくともそれらが異なるコピーであることを示すために異なる符号を付けている。ユーザは、キーボード162、および一般にマウス、トラックボール、またはタッチパッドと呼ばれるポインティングデバイス161などの入力デバイスを介してコンピュータ110にコマンドおよび情報を入力することができる。他の入力デバイス（図示せず）には、マイクロホン、ジョイスティック、ゲームパッド、サテライトディッシュ、スキャナ等が含まれる可能性がある。以上の入力デバイスおよび他の入力デバイスは、しばしば、システムバスに結合されたユーザ入力インターフェース160を介して処理ユニット120に接続されるが、パラレルポート、ゲームポート、またはユニバーサルシリアルバス（Universal Serial Bus）（USB）などの他のインターフェースおよびバス構造で接続してもよい。また、モニタ191または他のタイプの表示デバイスも、ビデオインターフェース190などのインターフェースを介してシステムバス121に接続される。モニタに加えて、コンピュータは、出力周辺インターフェース195を介して接続することができるスピーカ197やプリンタ196などの他の周辺出力デバイスも含むことが可能である。

## 【0027】

コンピュータ110は、遠隔コンピュータ180のような1つまたは複数の遠隔コンピュータに対する論理接続を使用するネットワーク化された環境で動作することも可能である。遠隔コンピュータ180は、他のパーソナルコンピュータ、サーバ、ルータ、ネットワークPC、ピアデバイス、または他の一般的なネットワークノードであることが可能であり、通常、パーソナルコンピュータ110に関連して前述した要素の多く、またはすべてを含むが、メモリ記憶デバイス181だけを図1に示している。図1に示した論理接続は、ローカルエリアネットワーク(LAN)171およびワイドエリアネットワーク(WAN)173を含むが、他のネットワークを含むことも可能である。そのようなネットワーク環境は、オフィス、企業全体のコンピュータ網、イントラネット、およびインターネットで一般的である。

## 【0028】

LANネットワーク環境で使用されるとき、パーソナルコンピュータ110は、ネットワークインターフェースまたはネットワークアダプタ170を介してLAN171に接続される。WANネットワーク環境で使用されるとき、コンピュータ110は、通常、インターネットなどのWAN173を介して通信を確立するためのモデム172、または他の手段を含む。内部にあることも、外部にあることも可能なモデム172は、ユーザ入力インターフェース160または他の適切な機構を介してシステムバス121に接続することが可能である。ネットワーク化された環境では、パーソナルコンピュータ110に関連して描いたプログラムモジュール、またはプログラムモジュールの部分を遠隔のメモリ記憶デバイスの中に記憶することができる。例として、限定としてではなく、図1は、遠隔アプリケーションプログラム185をメモリデバイス181上に常駐するものとして示している。図示したネットワーク接続は、例示的なものであり、コンピュータ間で通信リンクを確立する他の手段を使用してもよいことが認められよう。

## 【0029】

以下の説明では、特に明示しない限り、本発明を1つまたは複数のコンピュータによって行われる動作の処理および記号表現に関連して説明する。このため、ときとして、コンピュータによって実行されるものとして述べるそのような処理および動作には、コンピュータの処理ユニットによる構造化された形態のデータを表わす電気信号の操作が含まれるものと理解されたい。この操作により、データが変形されるか、またはコンピュータのメモリシステム内の場所に保持され、コンピュータの動作が、当分野の技術者によってよく理解されている形で再構成されるか、または他の方法で変更される。データが保持されているデータ構造は、データの形式によって定義された特定のプロパティを有するメモリの物理的な場所である。以上、本発明について説明したが、以下に説明する様々な処理および動作をハードウェアで実施することも可能である、しかしながら、本発明は、以下に示す実施例に限定されるものではない。

## 【0030】

図2は、本発明を実施する例示的な動作構成を示す図である。この動作構成は、ソース記憶媒体204および宛先記憶媒体206とインターフェースをとるコンピュータ200を含む。オペレーティングシステム202が、コンピュータ200内で実行される。オペレーティングシステム202は、コンピュータ200と記憶媒体204および206の間の通信を円滑にするファイルシステムドライバ208、イメージドライバ210、およびデバイスドライバ212などの構成要素を含む。

## 【0031】

各記憶媒体は、1つまたは複数のボリュームに論理的に分割される。記憶媒体204は、第1のボリューム214および第2のボリューム216に分割される。記憶媒体206は、単一のボリュームに分割される。各記憶媒体ボリュームには、オペレーティングシステム202がアクセス可能である。

## 【0032】

記憶媒体204および206はそれぞれ、ジオメトリを含む。記憶媒体のジオメトリには

、シリンダ ( cylinder ) の数、シリンダ当りのトラック数、およびトラック当りのセクタ数が含まれる。セクタは記憶媒体上の物理ユニットである。各セクタは、ある量の情報、例えば、512バイトを記憶することができる。各記憶媒体204および206のジオメトリは、同じである必要はない。

#### 【0033】

クラスタは、1つまたは複数のセクタのグループである。オペレーティングシステム202が、より詳細には、ファイルシステムドライバ208が、クラスタ番号を特定することによって読取り要求および書込み要求をデバイスドライバに送信する。ファイルシステムは、クラスタを構成するセクタの数を決定する。ファイルアロケーションテーブル16 ( FAT16 )、ファイルアロケーションテーブル32 ( FAT32 )、コンパクトディスクファイルシステム ( CDFS )、およびニューテクノロジーファイルシステム ( NTFS ) などの様々な周知のファイルシステムが存在する。

#### 【0034】

図示するとおり、ソース記憶媒体204の第1のボリューム214は、ソースデータ218を含む。ソースデータ218は、オペレーティングシステムファイル236、アプリケーションプログラムなどの実行可能プログラム238、およびアプリケーションプログラムで使用されるデータファイル240などの複数のファイルを含む。また、ソースデータ218は、制御情報242も含む。制御情報は、各ファイルに関するデータを含む記憶媒体ボリューム214上のクラスタを特定するファイル割振りデータ249を含む。また、制御情報242は、ビットマスク248も含む。ビットマスク248は、複数のビットを含み、各ビットが、記憶媒体ボリューム上の1つのクラスタに対応する。ビットマスク内の各ビットは、対応するクラスタが少なくとも1つのファイルに関する記憶されたデータを含むか、またはそのクラスタが割り振られていないままであるかを明らかにする値を有する。

#### 【0035】

制御情報242、およびファイル236、238、240を編成し、記憶するのに使用されるやり方は、使用されるファイルシステムに依存する。例として、NTFSファイルシステムは、マスタファイルテーブル ( MFT ) を記憶媒体上に記憶する。MFTは、記憶媒体上に記憶された各ファイルに関する範囲リストを含む。範囲リストは、開始ブロックおよび長さを明らかにする一連のエントリを含む。開始ブロックおよび長さは、ファイルに関するデータを含む一続きの連続するクラスタを定義する。

#### 【0036】

ソースデータ218の一部として含まれるその他の情報には、ページメモリ ( paged memory ) 244およびBIOSパラメータブロック ( BPB ) が含まれる。ページメモリ244は、記憶媒体に書き込まれるコンピュータのメモリの一部分である。BPBは、媒体のサイズや記憶媒体上のあるファイルの場所などの記憶媒体ボリュームに関するデータを含む。

#### 【0037】

宛先記憶媒体206は、イメージファイル220を含む。イメージファイルは、ソースデータ218を含む情報の論理的キャプチャである。本発明において、イメージファイル220は、ソースデータ218を記憶媒体ボリュームに復元するのに使用される。例えば、イメージファイル220は、記憶媒体ボリューム216上に復元データ215を生成するのに使用される。復元データ215には、元のソースデータ218のオペレーティングシステムファイル236、実行可能プログラム238、データファイル240、および制御情報242が含まれる。以下により詳細に説明するとおり、復元データ215内のデータのクラスタ域は、ソースデータ218からのデータのクラスタ域に直接に対応する必要はない。例えば、ソースデータ218は、イメージファイル220が作成されたときにデフラグされることが可能である。復元データ215をイメージファイル220の作成の元とされたソースボリューム214上に置き、これにより、ソース記憶媒体ボリューム214を以前の状態に復元することも可能である。

## 【0038】

また、イメージファイル220は、別個のコンピュータとインターフェースをとる記憶媒体上にソースデータ218を置くのにも使用される。イメージファイル220をLAN、WAN、またはインターネットなどのコンピュータ網を介して遠隔の記憶媒体上に復元することが可能である。また、イメージファイル220は、イメージファイル220がCD-ROMディスクまたはDVDディスクなどの取外し可能な媒体上に置かれる場合、別個のコンピュータに復元することも可能である。取外し可能な媒体が、別個のコンピュータとインターフェースをとり、その別個のコンピュータとインターフェースをとる別の記憶媒体ボリュームに復元される。

## 【0039】

宛先記憶媒体206は、イメージファイル220a、220b、220cも含む。本発明は、イメージファイル220の編集を可能にする。イメージファイル220は、編集された後、イメージファイル220aのような別個のファイルとして記憶されることが可能である。したがって、イメージファイルの複数のバージョンが作成される。代替の実施形態では、イメージファイル220は、第1の期間におけるソースデータ218を含む親イメージファイルである。イメージファイル220a、220b、220cは、後の期間に作成された子イメージファイルである。各子イメージファイル220a、220b、220cは、親イメージファイル220の作成の後にソースデータ218に加えられた変更だけを含む。他の方法では、子イメージファイル220a、220b、220cは、イメージファイル220の一部として記憶され、これにより、複数のバージョンのソースデータ218を有する単一のイメージファイルが作成される。

## 【0040】

イメージファイル220は、複数のイメージストリームを含む。図示するとおり、イメージファイル220は、制御ストリーム222、データストリーム224、ビットマップストリーム226、クラスタマップストリーム228、監査追跡ストリーム(audit trail stream)230、およびプロパティストリーム232を含む。イメージストリームは、イメージファイル220の一部として宛先記憶媒体206上に記憶される。また、イメージファイル220は、修正ストリーム234も含む。修正ストリーム234は、その他のイメージストリーム内のデータがイメージファイル220に書き込まれる前にそのデータを調整するのに使用される一時的なストリームである。その他のストリーム内のデータが調整されると、修正ストリーム234は、イメージファイルから取り除かれる。本発明の実施形態では、監査追跡ストリーム230およびプロパティストリーム232はオプションである。イメージファイル220a、220b、220cは、イメージファイル220と同じイメージストリームを含む。

## 【0041】

図示した実施形態では、宛先記憶媒体206およびソース記憶媒体214は、共通のデバイスドライバ212およびファイルシステムドライバ208を介してコンピュータ200とインターフェースをとっており、両方の記憶媒体は、共通のファイルシステムおよび同一の記憶媒体タイプを使用しているものと想定される。当分野の技術者には認められるとおり、異なる記憶媒体タイプを使用してもよく、また各記憶媒体ボリュームが、異なるファイルシステムを利用してもよい。例えば、記憶媒体ボリューム206が、CDFSファイルシステムを使用するCD-ROMであり、他方、記憶媒体ボリューム214が、FAT16またはFAT32を使用するハードディスクである。その場合、複数のファイルシステムドライバ208および複数のデバイスドライバ212が、通常、必要とされる。

## 【0042】

動作構成は、マウントツール250、キャプチャツール252、復元ツール254、およびアプリケーション256を含む。アプリケーション256は、例えば、ワードプロセッシングアプリケーションプログラム、データベースプログラム、および記憶媒体を管理するのに使用されるようなその他のシステムリソースを含む。記憶媒体を管理するのに使用される周知のリソースにより、ユーザが、記憶媒体上のファイルを閲覧すること、ファイ

ルをコピーすること、ファイルを削除すること、ファイルを移動すること等が可能になる。キャプチャツール252は、ソースデータ218からイメージファイル220を作成する。復元ツール254は、イメージファイル220から、元のソース記憶媒体ボリューム214などの記憶媒体ボリューム上、またはボリューム216などの別個の記憶媒体ボリューム上に復元データ215を作成する。

#### 【0043】

本発明の実施形態では、マウントツール250、キャプチャツール252、および復元ツール254は、独立型アプリケーションとして実装される。その場合、以上のツールは、ユーザインターフェース、アプリケーションプログラム、その他のプログラムモジュール、またはコマンドライン要求を介して実行されることが可能である。本発明の別の実施形態では、マウントツール250、キャプチャツール252、および復元ツール254は、ユーザインターフェース、アプリケーションプログラム、またはその他のプログラムモジュールの一部として実装される。

#### 【0044】

ツール250、252、および254は、要求をオペレーティングシステム202に送る。他の方法では、ツール250、252、254からの要求は、インターフェース231を介して経路指定される。インターフェース231が、イメージファイル220内の個々のストリーム222〜234内の情報に対するアクセスを提供する。したがって、拡張可能な1組のツールが提供される。すなわち、イメージファイル220を操作する新しいツールを作成することができる。

#### 【0045】

マウントツール250は、あたかもイメージファイル220が記憶媒体ボリュームであるかのように、オペレーティングシステム202がイメージファイル220と対話することを可能にするシーケンスを開始するのに使用される。マウントツール250は、イメージドライバ210やファイルシステムドライバ208などのオペレーティングシステム構成要素と協働して、イメージファイル220に関するアドレス指定可能なドライブ名マウントポイントを提供し、これにより、仮想記憶媒体を生成する。マウントツール250が実行された後、オペレーティングシステムは、イメージファイル220に対してデータの読取りおよび書き込みを行うことができる。マウントツールは、アプリケーション、ユーザインターフェース、システムリソース、またはコマンドライン要求によって開始される。例えば、マウントツールは、ブートシーケンス中に開始され、コンピュータ200が、イメージファイル220からブートすることが可能になる。

#### 【0046】

図3は、本発明によって使用されるイメージファイルの実施形態を示す図である。イメージファイル220は、制御ストリーム222、データストリーム224、ビットマップストリーム226、クラスタマップストリーム228、監査追跡ストリーム230、プロパティストリーム、および修正ストリーム234を含む。前述したとおり、監査追跡ストリーム230およびプロパティストリーム232は、オプションであり、さらなる機能をイメージファイル220に提供するために実装される。修正ストリーム234は、一時的であり、その他のイメージファイルストリームが宛先記憶媒体206上のイメージファイル220に書き込まれる前にそのストリームを変更するのに使用される。

#### 【0047】

制御ストリーム222は、イメージファイル220内に含まれるその他のデータを開き、解釈するのに必要とされるイメージファイル220の属性を記述するヘッダとして機能するデータ構造である。制御ストリーム内のデータは、イメージ識別260、ボリューム情報262、記憶媒体ジオメトリ264、ファイルシステム情報266、オペレーティングシステム情報268、および圧縮情報270を含む。

#### 【0048】

イメージ識別260は、イメージを一意的に特定するデータであり、また制御ストリーム222のバイトサイズを明らかにするデータでもある。ボリューム情報262は、ソース

記憶媒体ボリューム214を定義するデータを含む。ボリューム情報262内のデータは、記憶媒体204の先頭からのボリュームオフセット、ボリュームの長さ、ボリューム上の隠れたセクタの数、ボリューム番号、およびボリュームによって使用されるファイルシステムのタイプを含む。

【0049】

記憶媒体ジオメトリ264は、シリンダの数、シリンダ当たりのトラック数、およびトラック当たりのセクタ数を含め、ソース記憶媒体264の物理的属性を記述するデータを含む。ファイルシステム情報266は、クラスタの総数、クラスタ当たりのバイト数、およびレコード当たりのクラスタ数を明らかにするデータを含む。

【0050】

オペレーティングシステム情報268は、ソースデータ218の一部として記憶されたオペレーティングシステムバージョンを特定するデータを含む。また、オペレーティングシステム情報268は、サービスパック識別などのオペレーティングシステムに対して行われたあらゆる更新を明らかにするデータも含む。サービスパックは、ソフトウェアプログラムが抱える既知の問題を修正する、またはソフトウェアプログラムに拡張を提供するソフトウェアプログラムに対する更新である。サービスパック識別は、オペレーティングシステムファイル236に関してソース記憶媒体ボリューム214上に記憶されたあらゆるサービスパックを明らかにする。圧縮／暗号化情報は、イメージファイルが圧縮されているかどうか、またイメージファイル220が暗号化されているかどうかを明らかにするデータを含む。

【0051】

データストリーム224は、ソース記憶媒体ボリューム214からのソースデータ218を含む。前述したとおり、ソースデータ218は、ソース記憶媒体ボリューム214上の記憶されたファイルに関するデータを含む。各ファイルに関するデータは、オプションとして、周知の方法を使用して圧縮されるか、または暗号化される。データを圧縮または暗号化する決定は、ファイルごとに行われる。したがって、データストリーム224内の一部のデータは圧縮されるが、他のデータは圧縮されない可能性がある。同様に、データストリーム224内の一部のデータは暗号化されるが、他のデータは暗号化されない可能性がある。ソースデータ218の一部を成す制御情報242もファイルを構成し、データストリーム224の一部として記憶される。

【0052】

本発明の実施形態では、データストリーム224は、同一のデータの複数のコピーは含まない。例えば、同一のデータファイル240の複数のコピーがソース記憶媒体ボリューム214上に記憶されている場合、そのデータファイル240の1つのコピーだけが、データストリーム224の中に記憶される。また、ページメモリ244も、データストリーム224の中に含める必要がない。イメージファイル220が記憶媒体ボリューム上で復元される際、オペレーティングシステム202が、必要に応じて記憶媒体ボリューム上にページメモリ244を生成する。データファイルおよびページメモリファイルの複数のコピーをなくすことにより、イメージファイル220のサイズが小さくなる。

【0053】

クラスタマップストリーム228は、複数のレコードを含み、各レコードが、仮想クラスタ域およびデータストリーム域を特定する。各レコード内の仮想クラスタ域は、開始クラスタ272および終了クラスタ274を含む。開始クラスタ272および終了クラスタ274により、記憶媒体ボリュームに復元されたときに1組のソースデータが属する1組の連続するクラスタが特定される。データストリーム域は、オフセット278、長さ276、および圧縮サイズ280を含み、データストリーム224内でどこにそのデータセットが存在するかを明らかにする。オフセット278は、データストリームの先頭から計測されたデータストリーム224内のデータの開始位置を明らかにする。圧縮ファイルの場合、イメージファイル220内のデータセットの長さは、圧縮サイズ280に等しい。圧縮されていないデータの場合、イメージファイル220内のデータセットの長さは、制御ス



トリーム222内で明らかにされる長さ276とクラスタ当たりのバイト数の積である。フラグフィールド282は、イメージファイル220内のデータが圧縮されているか、または圧縮されていないかを明らかにする。クラスタマップストリーム228内のレコードにより、イメージファイル220を仮想記憶媒体ボリュームと見なすことが可能になり、またデータストリーム224内のデータを読み取ること、編集すること、または他の方法で操作することが可能になる。

#### 【0054】

図4は、データストリーム内のデータの位置がどのように仮想クラスタ域に相関するかの例を示す図である。仮想クラスタ域は、イメージファイルがボリューム上に復元される際に記憶媒体上でデータが置かれるクラスタに相当する。図4で、クラスタ当たりのバイト数は、制御ストリーム222内で特定される512であるものと想定している。仮想記憶媒体346が、順次に番号が付けられたブロックを含む。各ブロックが1つのクラスタを表わす。また、データストリーム224は、順次に番号が付けられたブロックも含む。データストリーム224内の各ブロックが、データストリーム224内のバイトを表わし、その番号が、データストリーム224の先頭からのバイト数でのオフセットを表わす。

#### 【0055】

クラスタマップレコード228の開始クラスタフィールド262は、開始クラスタ100を含み、また終了クラスタフィールド149は、終了クラスタ149を含む。開始クラスタおよび終了クラスタは、イメージファイルが記憶媒体上に復元される際に1組のデータが置かれる1組の連続するブロック、すなわち、仮想クラスタ域を定義する。オフセットフィールドは、1000という値を有する。したがって、データは、データストリーム224の先頭から1000バイトのオフセットで開始する。フィールド272は、データが圧縮されていないことを示している。したがって、データストリーム224内のこのデータの長さは、長さフィールド266の中で50と特定されている長さと、クラスタ当たりのバイト数512の積（50クラスタ×クラスタ当たり512バイト＝25600バイト）である。図示するとおり、イメージファイル内の対応するデータは、データストリーム224の先頭を起点としてバイト1000ないし26599に位置する。

#### 【0056】

前述したとおり、ソースデータ218は、イメージファイル220が作成されたときにデフラグされることが可能である。その場合、開始クラスタおよび終了クラスタは、ソース記憶媒体ボリューム214上の元の場所に対応しない可能性がある。代わりに、開始クラスタ番号および終了クラスタ番号は、仮想記憶媒体346によって表わされる、記憶媒体ボリュームにイメージファイル220が復元された際にデータが記憶されるクラスタを特定する。

#### 【0057】

同一のデータの複数のコピーがソース記憶媒体ボリューム214上に記憶されている場合、そのデータは、一度限りデータストリーム224の中に入れられる。これが行われた場合、クラスタストリーム228内で複数のレコードが生成され、ソース記憶媒体ボリューム214上でデータが出現する各回に関して1つのレコードが生成される。各レコードは、同じ長さフィールド266、同じオフセットフィールド262、同じ圧縮サイズフィールド270、および同じ圧縮フィールド272を含み、これにより、データストリーム224内の同一のデータをポイントする。開始クラスタフィールド262および終了クラスタフィールド264は異なり、これにより、イメージファイルが記憶媒体ボリュームに復元される際にデータが記憶される複数の場所が特定される。

#### 【0058】

図3に戻ると、ビットマップストリーム226を使用して、仮想割り振り済みクラスタおよび仮想未割り振りクラスタが特定される。仮想割り振り済みクラスタは、イメージファイルが記憶媒体ボリュームに復元された際にデータを含むクラスタである。仮想未割り振りクラスタは、イメージファイルが記憶媒体ボリュームに復元された際に新しいデータを記憶するのに利用可能なクラスタを表わす。図示した例では、ビットマップストリームは、ビット

マップとして実装される。ビットマップ内の各ビットが、クラスタに対応する。「1」という値を有するビットは、仮想割振りクラスタを表わし、他方、「0」という値を有するビットは、仮想未割振りクラスタを表わす。ビットマップストリームは、ビットマスク240と同じ機能を果たす。ただし、ビットマップストリーム226は、データストリーム224の外にあるため、イメージファイル220内で容易にアクセス可能である。

#### 【0059】

監査追跡ストリーム230は、イメージファイルに加えられたあらゆる変更、その変更を開始したユーザ、変更されているものがある場合、そのファイル、ならびにその変更が実施された日付および時刻を明らかにするデータを含む。プロパティストリーム232は、オペレーティングシステムバージョンなどのイメージファイル220の属性を明らかにするデータを含む。プロパティストリーム232には、所望の属性を有するイメージファイル220を特定するためにクエリが行われる。例えば、プロパティストリーム232にクエリが行われて、特定のバージョンのオペレーティングシステムを含むイメージファイルが探し出される。監査追跡ストリームおよびプロパティストリームは、任意の適切な仕方で実装される。

#### 【0060】

修正ストリームは、ソースデータ218がオペレーティングシステム202によって読み取られた後、複数のイメージストリームがイメージファイル220に書き込まれる前にそのストリームを変更するのに使用される。例えば、NTFSファイルシステムの場合、修正ストリームは、イメージファイルが作成された際にファイルがデフラグされる場合、MFTを変更する。

#### 【0061】

図5は、ソース記憶媒体ボリュームからイメージファイルを作成するのに使用されるプロセスを示す図である。プロセス290は、イメージドライバ210内で動作するプログラムモジュールによるなどして、オペレーティングシステム202によって実行されることが可能である。他の方法では、プロセス290は、キャプチャツール252などのシステムツールによって完全に、または部分的に実行されることが可能である。

#### 【0062】

通常、イメージファイル220は、システムツールからのユーザ要求を介して作成される。前述したとおり、システムツールは、ユーザインターフェース、アプリケーション、またはその他のプログラムモジュールの一部を介して実行されるか、または一部として実装される。また、システムツールは、コマンドライン要求からも実行される。要求がシステムツールから受け取られた後、オペレーティングシステム202が、ステップ292に示すとおり、ソース記憶媒体ボリューム214を開く。ソース記憶媒体ボリューム214を開くのに、イメージドライバ210は、ファイルシステムドライバ208を介してソース記憶媒体ボリューム214に対するリファレンスを獲得する。ソース記憶媒体ボリューム214に対するリファレンスが獲得されると、オペレーティングシステム202は、イメージドライバ210およびデバイスドライバ212を介して記憶媒体214と通信する。

#### 【0063】

ステップ294で、プロセス290は、制御ストリーム222、データストリーム224、ビットマップストリーム226、クラスタマップストリーム228、監査追跡ストリーム230、プロパティストリーム232、および修正ストリーム234を含むイメージファイル220の複数のイメージストリームを生成する。この時点で、イメージファイル220を構成する複数のストリームをメモリ構造内におけるのと同様に実装することが可能であり、あるいは、宛先記憶媒体ボリューム206上に直接に実装することが可能である。

#### 【0064】

ステップ296で、プロセス290が、ソース記憶媒体ボリューム214上の制御情報242を読み取る。制御情報242は、ソース記憶媒体ボリューム214上の複数のファイル、およびそのファイルのそれぞれに割り振られたクラスタを識別する。例えば、ソース

記憶媒体ボリュームがNTFSファイルシステムを使用する場合、プロセス290は、記憶媒体ボリューム214上のMFTを読み取る。

【0065】

ステップ298で、プロセス290は、制御情報242によって特定された第1のファイルに関するデータを読み取る。ステップ300で、プロセスは、そのファイルに関するデータをデータストリーム224の中に入れる。前述したとおり、そのファイルデータの複数のコピーが存在する場合、データは、データストリームの中に一度限り入れられる。さらに、ファイルデータがページメモリ244の一部である場合、そのデータは、データストリーム224の中に入れる必要がない。ファイルに関するデータは、オプションとして、データストリーム224の中に入れられる際にデフラグされる。

【0066】

ステップ302で、仮想クラスタ域をデータストリーム域に相関させるレコードを追加することにより、クラスタストリーム228が更新される。ステップ304で、プロセス290は、ソース記憶媒体ボリューム214がさらなるファイルデータを含むかどうかを判定する。さらなるファイルデータが存在する場合、プロセス290は、ステップ298に戻り、次のファイルに関するデータを読み取る。

【0067】

もはやファイルデータが存在しない場合、プロセス290は、ステップ306に進み、修正をイメージファイルストリームに適用する。この修正は、データの移動を考慮に入れるように適用される。例えば、前述したとおり、プロセス290は、ソース記憶媒体214上の複数のファイルに関するデータをデフラグする可能性がある。したがって、プロセス290は、データストリーム224内にある制御情報242を変更して、制御情報242が、各ファイルに現在、割り振られているクラスタを特定するようにする。NTFSボリュームの場合、プロセスは、MFT内の各ファイルに関する範囲リストを訂正する。各ファイルに関するデータは、デフラグされているため、各ファイルに関する範囲リストは、単一の範囲を含む。また、ビットマスク248も、データ位置の変更に鑑みて、割り振り済みのクラスタおよび未割り振りのクラスタを適切に反映するように変更される。また、修正は、BPB246にも適用される。BPB内で参照されるファイルの場所を変更されているため、BPBは、ファイルの適切な場所が参照されるように変更される。

【0068】

以上のプロセスが完了すると、制御ストリーム222、データストリーム224、ビットマップストリーム226、クラスタマップストリーム228、監査追跡ストリーム230、およびプロパティストリーム232は、宛先記憶媒体206に書き込まれ、これにより、ステップ308に示したイメージファイル220が形成される。

【0069】

図6は、ソース記憶媒体ボリュームなどの記憶媒体ボリュームにイメージファイルを復元するのに使用されるプロセスを示す図である。プロセス310は、オペレーティングシステム202によって実行されることが可能である。別法では、プロセス310は、復元ツール254などのシステムツールによって完全に、または部分的に実行される。ステップ312で、プロセス310は、イメージファイル220を開き、宛先記憶媒体ボリューム206に対するリファレンスを獲得する。

【0070】

ステップ314で、プロセス310は、クラスタマップストリーム228内に含まれる複数のクラスタマップレコードの1つを読み取る。クラスタマップレコードは、データに関するデータストリーム224内の場所を特定し、また、ソース記憶媒体ボリューム214上のそのデータの意図される場所、すなわち、仮想クラスタ域も特定する。クラスタマップレコードを読み取った後、ステップ316で示すとおり、データが、データストリーム224から読み取られる。次に、データは、記憶媒体214に、クラスタマップ228内の開始クラスタおよび終了クラスタによって特定されたクラスタ上で書き込まれる。

【0071】

ステップ320で、プロセス310は、クラスタマップストリーム228内でさらなるレコードが存在するかどうかを判定する。さらなるレコードが存在する場合、プロセスは、ステップ314に戻り、次のクラスタマップレコードが検査される。もはやレコードが存在しない場合、プロセス310は、ステップ322に進み、修正が適用される。この修正は、イメージファイルを作成するのに使用されたソース記憶媒体ボリューム214と、ソースデータ218が復元される記憶媒体ボリュームの間でジオメトリが異なる場合、必要とされる。例えば、BPB内で特定されたファイルの場所が変更されている場合、BPBが、ファイルのその新しい場所を反映するように更新される。また、BPBは、記憶媒体のジオメトリ情報も含む。BPB内のジオメトリ情報は、復元されたソースデータ218を含む記憶媒体の実際のジオメトリを反映するように更新される必要がある。

#### 【0072】

図7は、オペレーティングシステムによるイメージファイルへのアクセスを提供するのに使用されるプロセスを示す図である。プロセス328は、マウントツール250からのユーザ要求の結果として開始される。別法では、プロセス328は、アプリケーション256が、イメージファイル220にアクセスしようと試みたとき、またはコマンドライン要求が受け取られたときに開始される。ステップ330で、マウント要求が、イメージドライバ210によって受け取られる。

#### 【0073】

イメージドライバ210は、マウント要求を受け取った後、ステップ332に示すとおり、イメージファイル220を開く要求をファイルシステムドライバ208に送る。ファイルを開くことは、記憶媒体上のファイルに対するアクセスを許すためにオペレーティングシステム202によって使用される標準のプロセスである。イメージファイル220は、記憶媒体206上のファイルとして記憶されているため、アクセスするにはまず、開かなければならない。

#### 【0074】

イメージファイル220が開かれると、イメージドライバ210は、ファイルシステムドライバ208を迂回して、イメージファイル220を直接に読み取る。イメージドライバ210は、ステップ334および336に示すとおり、制御ストリーム222およびクラスタマップストリーム228を読み取る。前述したとおり、制御ストリームは、ソースデータ218を編成するのに使用されるファイルシステムのタイプを特定し、またソース記憶媒体214のジオメトリも特定するデータを含む。制御ストリーム内のデータは、データストリーム224などのその他のストリーム内の情報を解釈するのに必要である。クラスタマップストリーム228は、仮想クラスタ域をデータストリーム域に相関させるのに必要な情報を含む。

#### 【0075】

ステップ338で、イメージドライバ210は、イメージファイル220に対するドライブ名マウントポイント割当てを要求する。この割当ては、ドライブ名マウントポイントを割り当てることを担うオペレーティングシステム202の部分によって行われる。

#### 【0076】

図8は、どのようにイメージドライバがイメージファイルと協働して、イメージファイル内のデータから仮想記憶媒体ボリュームを生成するかを示す図である。イメージファイル220がマウントされると、ファイルシステムドライバ208は、あたかもイメージファイル220が記憶媒体ボリュームであるかのように、イメージファイル220にアクセスする。イメージドライバ210は、図示するとおり、ファイルシステムドライバ208の下で動作し、ファイルシステムドライバ208からのアクセス要求を変換する。簡明にするため、デバイスドライバ212は図示していない。

#### 【0077】

例として、ファイルシステムドライバ208は、イメージファイルからデータを読み取るデータ要求340を発行するとき、一群のクラスタを特定する。このクラスタは、復元データ215の中で所望のデータが存在する場所に対応する。したがって、オペレーティン

グシステム202は、実際には、一群の仮想クラスタを提供する。イメージドライバ210が、データ要求340を代行受信する。次に、イメージドライバは、クラスタマップストリーム228を検査し、ファイルシステムドライバ208によって提供された仮想クラスタに対応する開始クラスタおよび終了クラスタを有するレコードを探し出す。次に、イメージドライバは、クラスタマップレコードに基づいてデータストリーム内で所望のデータを探し出し、データストリームからそのデータを読み取り、所望のデータ342をファイルシステムドライバ208に転送する。

#### 【0078】

以上の構成により、あたかもイメージファイル220が記憶媒体ボリュームであるかのように、オペレーティングシステムが、イメージファイル220からデータを要求することが可能になる。したがって、オペレーティングシステム202は、データストリーム224内に記憶された制御情報242、データファイル240、実行可能プログラム238、およびその他のデータを読み取ることができる。この結果、オペレーティングシステム202は、イメージファイルを、346のラベルが一般的に付けられている仮想記憶媒体と見なす。アプリケーション256が、例えば、オペレーティングシステム202を介してイメージファイル220からのデータファイル240を読み取ることができる。次に、アプリケーション256は、そのデータファイル240を編集することができる。別の例として、記憶媒体のコンテンツを読み取り、表示するのに使用されるものなどの記憶媒体リソースが、イメージファイルに対して使用される。記憶媒体リソースは、あたかもイメージファイルが復元データ215であるかのように、そのイメージファイルのコンテンツを表示する。

#### 【0079】

イメージファイルのアクセスが許されるため、用意されたツールのセットは、拡張可能である。例えば、インターフェース231が、イメージファイル220内のデータストリームへのアクセスを可能にするアプリケーションプログラミングインターフェース231を形成することが可能である。したがって、このアーキテクチャにより、イメージファイル220を作成する、復元する、閲覧する、編集する、または他の方法で操作する新しいツールを生成することが可能になる。

#### 【0080】

また、この動作構成により、イメージファイル220を編集することが可能になる。ファイルシステムドライバ208が、データを一群のクラスタに書き込むデータ要求340を送る。この一群のクラスタは、1組の仮想未割り振りクラスタに対応する。イメージドライバ210が、その要求を代行受信し、248のラベルが付けられたデータをデータストリーム224の終端に書き込む。データが新しいファイルである場合、イメージファイルドライバ210は、仮想クラスタ域を特定する新しいクラスタマップレコードを生成する。クラスタマップレコードは、データストリーム域をさらに含む。

#### 【0081】

データが編集されたファイルである場合、イメージドライバは、そのデータを、248の番号が付けられたデータストリーム224の終端に書き込む。ただし、新しいクラスタマップ228レコードは、生成されない。代わりに、イメージドライバは、ファイルデータの以前の場所を特定するレコードを探し出し、新しい仮想クラスタ域および新しいデータストリーム域を含むようにそのレコードを変更する。

#### 【0082】

図9は、複数のバージョンのイメージファイルを示す図である。親イメージファイル220は、第1の期間におけるソースデータ218のキャプチャである。前述したとおり、制御ストリームは、イメージファイル220を一意的に特定するデータを含む。子イメージファイル220aは、第2の期間におけるソースデータ218のキャプチャである。ただし、子イメージファイル220aのデータストリームおよびクラスタマップストリームは、第1の期間から第2の期間までにソースデータ218に加えられたデータ変更だけを表わす。例えば、親イメージファイル220が作成された後に新しいオペレーティングシス

テムプログラム236が追加された場合、子イメージファイル220aは、その新しいファイルに関するデータだけを含む。データストリーム内およびディスク上でその新しいファイルデータの場所をサポートするクラスタマップレコードが、子ファイル220a内で生成される。子イメージファイル220aは、親イメージファイル220を特定するデータを制御ストリームの中に含める。子イメージファイル220cも同様に、第3の期間におけるソースデータ218を表わす。

【0083】

親イメージファイル220および子イメージファイル220a、220bは、別々のファイルとして記憶される。他の方法では、親ファイル220および子イメージファイル220a、220bは、より大きいファイル206の一部として記憶される。親イメージファイルおよび子イメージファイルを使用することの利点は、イメージファイル全体を再生成する必要なしに、複数のバージョンのイメージが保持されることである。したがって、複数のバージョンのイメージファイルの全体のサイズが小さくなる。

【0084】

図10は、ネットワークなどの媒体を介してイメージファイルを伝送する方法を示す図である。図示するとおり、イメージファイル220は、複数のセグメント360、362、364、366に細分される。各セグメントは、ネットワークを介して別々に伝送可能な1組のデータを表わす。第1のセグメント360は、制御ストリーム222、ビットマップストリーム226、プロパティストリーム232、監査追跡ストリーム230、修正ストリーム234、およびクラスタマップストリーム228を含む。また、第1のセグメントは、224aのラベルが付けられたデータストリームの一部分も含む。残りのセグメント362、363、364は、データストリームの他の部分、224b、224c、224dを含む。集合的に、部分224a、224b、224c、および224dは、データストリーム224全体を形成する。

【0085】

各セグメント360、362、363、364は、ネットワークを介して別々に伝送可能である。したがって、第1のコンピュータがイメージファイル220をネットワーク上の第2のコンピュータに送信し、プロセス中にネットワーク接続が失われた場合、第1のコンピュータは、第2のコンピュータによって完全に受信されなかったセグメントを再送信するだけでよい。

【0086】

本発明の原理を適用することが可能な多数の可能な実施形態に鑑みて、図面を参照して本明細書で説明した実施形態は、単に例示的なものに過ぎず、本発明の範囲を制限するものと受け取られるべきでないことを認識されたい。例えば、本発明の趣旨逸脱することなく、ソフトウェアで示される例示した実施形態の要素をハードウェアで実装することが可能であり、またその逆も可能であること、あるいは、例示した実施形態の構成および詳細を変更できることが、当分野の技術者には認められよう。したがって、本明細書で説明した本発明は、特許請求の範囲の範囲に入る可能性があるすべてのそのような実施形態、およびそれと等価な形態を含んでいる。

【図面の簡単な説明】

【図1】本発明が存在する例示的なコンピュータシステムを一般的に示すブロック図である。

【図2】イメージファイルの作成、復元、および編集を行うのに使用される動作環境を示すブロック図である。

【図3】本発明で使用されるイメージファイル形式を示すブロック図である。

【図4】本発明で使用されるクラスタストリームを示すブロック図である。

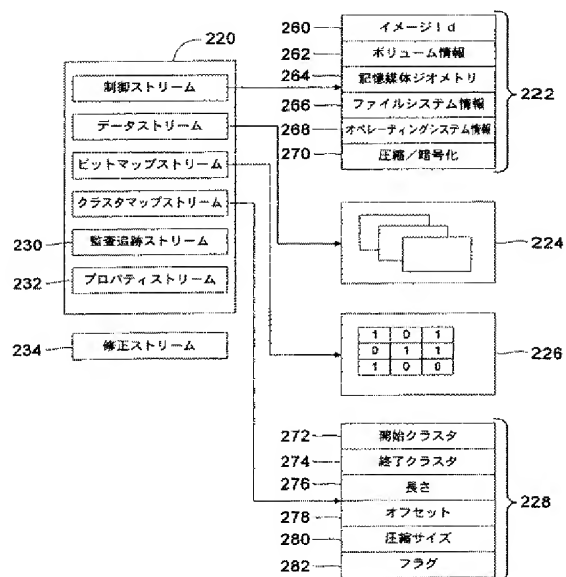
【図5】イメージファイルを作成するのに使用されるプロセスを示すブロック図である。

【図6】イメージファイルを復元するのに使用されるプロセスを示すブロック図である。

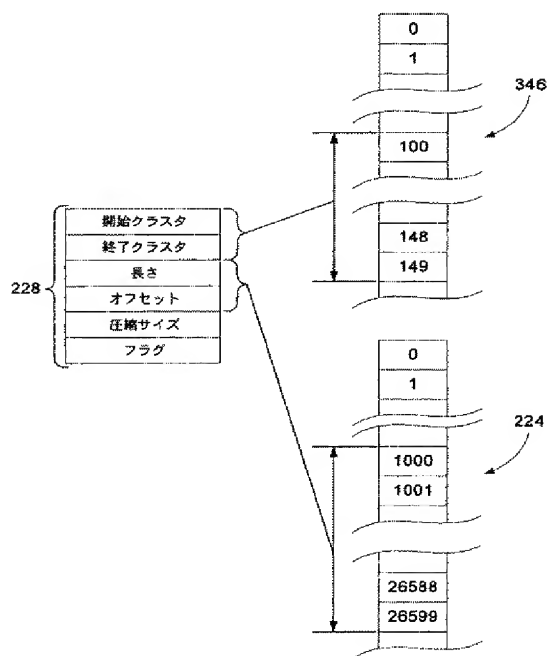
【図7】イメージファイル内のデータへのアクセスを可能にするのに使用されるプロセスを示すブロック図である。



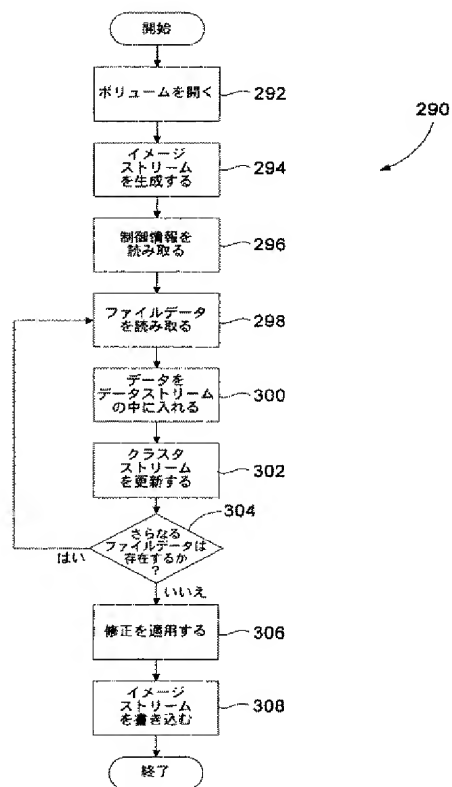
【図3】



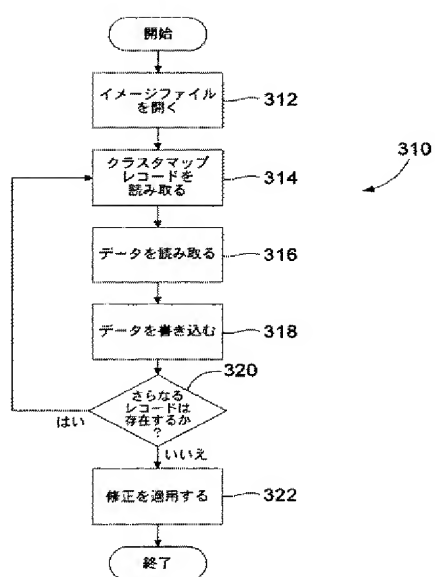
【図4】



【図5】

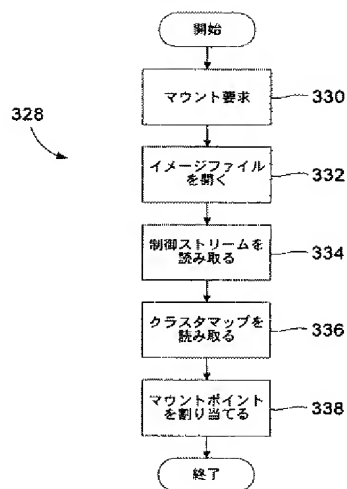


【図6】

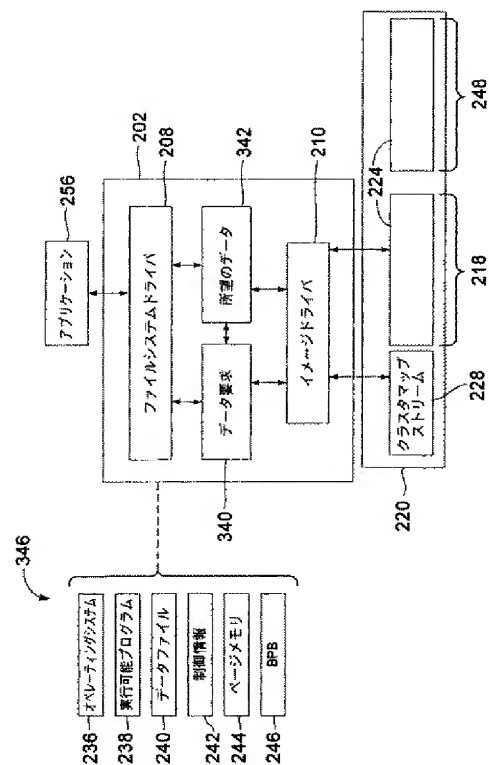




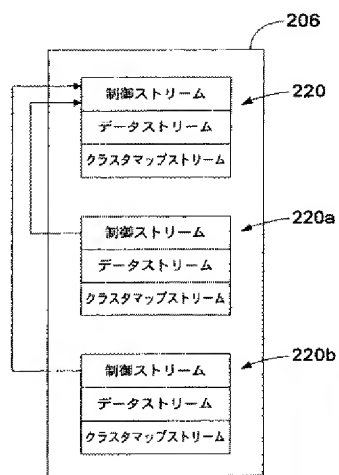
【図7】



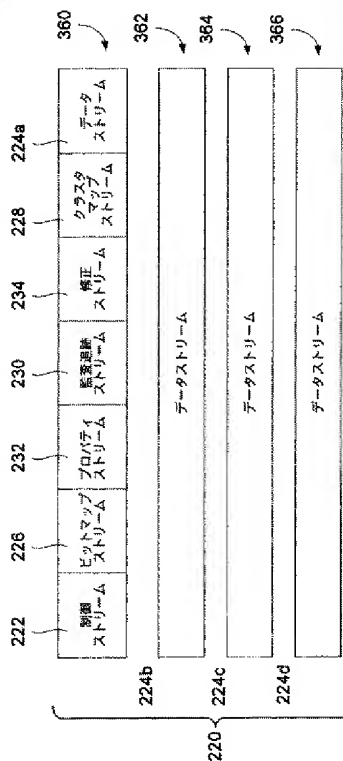
【図8】



【図9】



【図10】



(72)発明者 サラ ジェイ. シューマッハー

アメリカ合衆国 98102 ワシントン州 シアトル イースト ハリソン ストリート 10  
17 アpartment 205

(72)発明者 カーティク エヌ. ラガバン

アメリカ合衆国 98109 ワシントン州 シアトル デイラー ストリート 201 アパー  
トメント 511

Fターム(参考) 5B017 AA07 BA07 CA16

5B082 DE06 GA01 GA11 GA14